A hybrid metaheuristic approach to the university course timetabling problem

Abdullah, S., Turabieh, H., McCollum, B., & McMullan, P. (2012). A hybrid metaheuristic approach to the university course timetabling problem. *Journal of Heuristics*, *18*(1), 1-23. https://doi.org/10.1007/s10732-010-9154-y

Published in:
Journal of Heuristics

Document Version:
Publisher's PDF, also known as Version of record

J Heuristics (2012) 18:1–23
DOI 10.1007/s10732-010-9154-y

A hybrid metaheuristic approach to the university course timetabling problem

Salwani Abdullah · Hamza Turabieh · Barry McCollum · Paul McMullan

Received: 27 August 2009 / Revised: 3 November 2010 / Accepted: 14 December 2010 /
Published online: 24 December 2010

Abstract This paper describes the development of a novel metaheuristic that combines an electromagnetic-like mechanism (EM) and the great deluge algorithm (GD) for the University course timetabling problem. This well-known timetabling problem assigns lectures to specific numbers of timeslots and rooms maximizing the overall quality of the timetable while taking various constraints into account. EM is a population-based stochastic global optimization algorithm that is based on the theory of physics, simulating attraction and repulsion of sample points in moving toward optimality. GD is a local search procedure that allows worse solutions to be accepted based on some given upper boundary or 'level'. In this paper, the dynamic force calculated from the attraction-repulsion mechanism is used as a decreasing rate to update the 'level' within the search process. The proposed method has been applied to a range of benchmark university course timetabling test problems from the literature. Moreover, the viability of the method has been tested by comparing its results with other reported results from the literature, demonstrating that the method is able to produce improved solutions to those currently published. We believe this is due to the combination of both approaches and the ability of the resultant algorithm to converge all solutions at every search process.

S. Abdullah · H. Turabieh (✉)
Data Mining and Optimisation Research Group, Center for Artificial Intelligence Technology,
University Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia
e-mail: hamza@ftsm.ukm.my

S. Abdullah
e-mail: salwani@ftsm.ukm.my

B. McCollum · P. McMullan
Department of Computer Science, Queen's University Belfast, Belfast BT7 1NN, UK

B. McCollum
e-mail: b.mccollum@qub.ac.uk

P. McMullan
e-mail: p.mcmullan@qub.ac.uk

Keywords Dynamic force decay rate · Great deluge algorithm · Electromagnetism-like mechanism · Course timetabling

1 Introduction

Course timetabling is a multi-dimensional assignment problem in which students and faculty members are assigned to courses (or events), and courses are assigned to classrooms and timeslots. Course timetabling problems can be classified into two categories; post-enrollment and curriculum-based timetabling problems. The main difference is that the post-enrollment course timetabling problems concentrate on students' preferences' such as "a student should only have one course per day", while curriculum-based course timetabling focuses on lecturers' preferences such as "lecturer only wants their lectures in the morning". This area of research has long attracted the attention of the Operational Research and Artificial Intelligence communities. In addition, variations of the problem have been the subject of two competitions (http://www.metaheuristics.org and www.cs.qub.ac.uk/itc2007, McCollum et al. 2010). In the past, a wide variety of approaches for constructing course timetables have been described and discussed in the literature. Carter (1986) categorized these approaches into four types: sequential methods, cluster methods, constraint-based methods and generalized search. Petrovic and Burke (2004) added the following categories: hybrid evolutionary algorithms, meta-heuristics, multi- criteria approaches, case based reasoning techniques, hyper-heuristics and adaptive approaches.

In this paper, the discussion on the previous work on the course timetabling problem is divided into two prevalent groups i.e. (i) algorithms in solving enrollment-based course timetabling problems and (ii) algorithms that are employed to tackle curriculum-based course timetabling problem. Full descriptions of the two types of problems are outlined in Sect. 2, highlighting the main differences between them.

1.1 Enrollment-based course timetabling problems

In 2002, the International Metaheuristic Network organized the First International Timetabling Competition (ITC2002). This used artificially generated enrollment-based course timetabling problems which have become a standard within the research area, used within novel techniques being trialed by many researchers within their work. The details of these instances are available at http://www.idsia.ch/Files/ttcomp2002.

Using the same data generator, Socha et al. (2002) introduced a further eleven course timetabling problem data sets, and applied local search and ant based approaches to the problem. Rossi-Doria et al. (2003) considered evolutionary algorithms for the same datasets and presented a comparison of a number of metaheuristic methods. Burke et al. (2003) introduced a tabu-based hyperheuristic and applied it to university course timetabling in addition to nurse rostering. Burke et al. (2007) employed tabu search within a graph based hyper-heuristic and applied it to both examination and course timetabling benchmark datasets with the aim of raising the level of generality by operating on different problem domains. Abdullah et al. (2005)

developed a variable neighborhood search approach which used a fixed tabu list to penalize particular neighborhood structures. Abdullah et al. (2007a) applied a randomized iterative improvement approach using a composite of eleven neighborhood structures. Abdullah et al. (2007b) combined this with a mutation operator, introducing a hybrid approach to the problem. Asmuni et al. (2005) employed a fuzzy method to order three different heuristics. McMullan (2007) applied the extended great deluge algorithm to the datasets introduced by Socha et al. (2002). Landa-Silva and Obit (2008) introduced non-linear great deluge which generates non-linear decay rate for three different categories of datasets. The combination of genetic algorithm and local search has been employed by Abdullah and Turabieh (2008) and is able to produce promising results on the same test instances.

1.2 Curriculum-based course timetabling problems

Curriculum-based course timetabling was first introduced in The 2nd International Timetabling Competition (ITC2007) by Gaspero et al. (2007). ITC2007 consists of three tracks representing educational timetabling problems. Track 1 represents an examination timetabling problem; Track 2 represents a post-enrollment case study which focuses on students enrollment for courses; and Track 3 represents a curriculum-based course timetabling problem which focused on lecturers' preferences rather than students' preferences (as in Track 2). There are a number of papers available in the literature that focus on curriculum-based course timetabling. Müller (2007) applied a constraint-based solver to the curriculum-based course timetabling problems in the 2nd International Timetabling Competition, ITC2007 (Track 1 and Track 3) and achieved the first place in this competition. Lü and Hao (2010) applied a hybrid heuristic algorithm called Adaptive Tabu Search (ATS) to the same instances. Clark et al. (2008) applied repair-based heuristic search on Track 3 datasets in the ITC2007 timetabling competition. Geiger (2008) applied a stochastic neighborhood method based on threshold acceptance criteria to overcome the local optima to the same instances. Atsuta et al. (2007) applied the constraint satisfaction problem (CSP) which implemented a hybridization of tabu search and iterated local search algorithms to handle weighted constraints. This solver has been applied on Track 1, Track 2 and Track 3 in ITC2007.

De Cesco et al. (2008) applied a dynamic tabu search to curriculum-based course timetabling, a short term tabu exclusion with variable size tabu length, with dynamic weight adjustment for hard and soft constraints. Lach and Lübbecke (2008) applied an integer programming method on the same instances. Burke et al. (2009) introduced a new solver based on a hybrid metaheuristic to tackle scheduling problems. They applied it first on Udine data sets (based on Track 3 of ITC2007), achieving good solutions within a practical timeframe. Lü and Hao (2010) reported that the results obtained by Schaerf were from the employment of a tabu search with a relaxed stopping condition.

A survey of practical approaches to the problem, up to 1998, can be seen in Carter and Laporte (1996). The following papers represent a comprehensive list of surveys and overviews on educational timetabling (which include issues related to university course timetabling) i.e. Bardadym (1996), Burke and Petrovic (2002), Burke et al.

(1996), Carter (1986), Petrovic and Burke (2004), Schaerf (1999), de Werra (1985) and McCollum (2007) that discussed issues of bridging the gap between theory and practice in university timetabling.

The paper is organized as follows: The next section describes the post-enrollment and curriculum-based course timetabling problem. Section 3 describes our algorithm and its application to the course timetabling problem. Experimental results are discussed in Sect. 4. Some brief concluding comments are presented in Sect. 5.

2 Problem description

2.1 Problem A: post-enrollment-based course timetabling

The problem involves the assignment of lecture events to timeslots and rooms subject to a variety of hard and soft constraints. Hard constraints represent an absolute requirement. A timetable which satisfies the hard constraints is known as a *feasible* solution. The problem description that is employed in this paper is adapted from the description presented in Socha et al. (2002) and was the same as the description used in the first international timetabling competition (ITC2002). The following hard and soft constraints are presented:

- No student can be assigned to more than one course at the same time.
- The room should satisfy the features required by the course.
- The number of students attending the course should be less than or equal to the capacity of the room.
- No more than one course is allowed in each room during each timeslot.

Soft constraints that are equally penalized are as follows:

- A student has a course scheduled in the last timeslot of the day.
- A student has more than 2 consecutive courses.
- A student has a single course on a day.

The problem has:

- A set of N courses, $e = \{e_1, \ldots, e_N\}$
- 45 timeslots
- A set of R rooms
- A set of F room features
- A set of M students.

The objective is to satisfy the hard constraints and to minimize the violation of the soft constraints. In real-world situations, it is usually impossible to satisfy all soft constraints, but minimizing the violations of soft constraints represents an increase in the quality of the solution.

The experiments for the standard course timetabling problem were tested on the benchmark course timetabling problems proposed by the Metaheuristics Network[1]

[1] http://www.metaheuristics.org/.

Table 1 Parameter values for the course timetabling problem categories (Socha datasets)

Category	Small	Medium	Large
Number of events	100	400	400
Number of rooms	5	10	10
Number of features	5	5	10
Number of students	80	200	400
Maximum courses per student	20	20	20
Maximum student per courses	20	50	100
Approximation features per room	3	3	5
Percentage feature use	70	80	90

that involve scheduling 100–400 events/courses into a timetable with 45 timeslots corresponding to 5 days of 9 hours each, whilst satisfying room features and room capacity constraints. In this work, we considered two cases of post-enrollment course timetabling problems i.e. (i) the first case was proposed by Socha et al. (2002) and (ii) the second case was proposed by the first international timetabling competition (ITC2002).

Socha et al. (2002) datasets are divided into three categories: small, medium and large. We deal with 11 instances: 5 small, 5 medium and 1 large. The parameter values defining the categories are given in Table 1. These parameters consists of number of events, rooms, features, students, maximum number of courses for each student, maximum number of students for each course, approximation (average) features for each room and percentage of features used in each category.

The first international timetabling competition (ITC2002) introduced 20 instances. The parameter values defining the categories are given in Table 2.

2.2 Problem B: curriculum-based course timetabling

The curriculum-based course timetabling problem deals with the weekly assignment of a set of lectures for several university courses to specific timeslots and rooms, where conflicts between courses are set according to curricula published by the university and not on the basis of enrollment data. The curriculum-based course timetabling problem is considered as the third track in the 2nd International timetabling competition (ITC2007). The main reason for wide acceptance of this formulation is that it can represent real problems that often arise in real higher educational institutions. In this paper, we consider the same curricula-based course timetabling problem as described in Gaspero et al. (2007) that consist of the following entities: Days, Timeslots and Periods. We are given a number of teaching days in the week from 5 to 6.

- Each day is divided into a fixed number of timeslots, which is equal for all days.
- A period is a pair composed of a day and a timeslot. The product of the days and the timeslots represent the total number of periods.
- Each course consists of a fixed number of lectures to be scheduled in distinct periods which is taught by a teacher and attended by a number of students. Each course

Table 2 Parameter values for the first international timetabling competition (ITC2002)

Instance Identifier	Number of events	Number of Students	Number of Rooms	Rooms/event	Events/ Students	Students/ Event
1	400	200	10	1.96	17.75	8.88
2	400	200	10	1.92	17.23	8.62
3	400	200	10	3.42	17.70	8.85
4	400	300	10	2.45	17.43	13.07
5	350	300	10	1.78	17.78	15.24
6	350	300	10	3.59	17.77	15.23
7	350	350	10	2.87	17.48	17.48
8	400	250	10	2.93	17.58	10.99
9	440	220	11	2.58	17.36	8.68
10	400	200	10	3.49	17.78	8.89
11	400	220	10	2.06	17.41	9.58
12	400	200	10	1.96	17.57	8.79
13	350	250	10	2.43	17.69	11.05
14	350	350	10	3.08	17.42	17.42
15	350	300	10	2.19	17.58	15.07
16	440	220	11	3.17	17.75	8.88
17	350	300	10	1.11	17.67	15.15
18	400	200	10	1.75	17.56	8.78
19	400	300	10	3.94	17.71	13.28
20	350	300	10	3.43	17.49	14.99

has a minimum number of days within which the lecture for that particular course should be spread. There are some periods in which the course cannot be scheduled.

- Each room has a capacity and location. Capacity is represented in terms of available seats. Location is represented as an integer value corresponding to a separate building. Some rooms are not suitable for some courses due a lack of required equipment.
- A curriculum represents a group of courses such that any pair of courses in the group has students in common. Based on curricula, we have the conflicts between courses and other soft constraints.

The solution of the problem is an assignment of a period (day and timeslot) and a room to all lectures of each course. The following hard and soft constraints are presented:

Hard constraint

There are four hard constraints considered in this paper as in Gaspero et al. (2007):

H1: *Lectures*: All lectures of a course must be scheduled, and assigned to distinct periods. A violation occurs if a lecture is not scheduled or two lectures within a course are scheduled in the same period.

H2: *Conflicts*: All lectures of courses in the same curriculum or taught by the same teacher must be scheduled in different periods. Two conflicting lectures in the same period represent one violation. Three conflicting lectures count as 3 violations: one for each pair.

H3: *Availability*: If the teacher of the course is not available to teach that course at a given period, then no lecture of the course can be scheduled at that period. Each lecture scheduled in a period unavailable to that course is one violation.

H4: *Room Occupation*: Two lectures cannot be assigned to the same room at the same period. Two lectures in the same room at the same period represent one violation. Any extra lecture in the same period and room counts as one more violation.

Soft Constraint

The soft constraints are also taken from Gaspero et al. (2007):

S1: Room Capacity: The number of students that attend the course for each lecture must be less than or equal to the number of seats of the rooms hosting its lectures. Each student above the capacity counts as 1 violation.

S2: Minimum Working Days: The lectures of each course must be spread over the given minimum number of days. Each day below the minimum, counts as 1 violation.

S3: Isolated Lectures: Lectures belonging to a curriculum should be adjacent to each other (i.e., in consecutive periods). For a given curriculum we account for a violation every time there is one lecture not adjacent to any other lecture within the same day. Each isolated lecture in a curriculum counts as 1 violation.

S4: Room stability. All lectures of a course should be delivered in the same room. Each distinct room used for the lectures counts as 1 violation.

The experiments for the curriculum-based course timetabling problem discussed in this paper were tested on the twenty one real-world instances provided by the University of Udine. The main features of the instances used are given in Table 3. The details of all instances can be found in http://tabu.diegm.uniud.it/ctt/index.php.

3 The algorithm

3.1 Hybrid construction heuristic

A construction algorithm proposed by Chiarandini et al. (2006) and Landa-Silva and Obit (2008) is used to generate large populations of random initial solutions. The construction algorithm consists of three phases as presented in pseudo code form in Fig. 1.

We called this construction a hybrid constructive algorithm. This approach was chosen in particular because it was able to produce feasible solutions for all datasets due to the combination of the strength from three phases involved (see Landa-Silva and Obit 2008).

3.1.1 Phase 1: Largest degree heuristic

In this phase, we start with an empty timetable. The courses with the largest number of conflicts are scheduled first. All courses are scheduled by randomly selecting

Table 3 Curriculum-based instances

Instance	Course	Total Lectures	Rooms	Period per Day	Days	Curricula	Min and Max Lectures per Days per Curriculum
comp01	30	160	6	6	5	14	2–5
comp02	82	283	16	5	5	70	2–4
comp03	72	251	16	5	5	68	2–4
comp04	79	286	18	5	5	57	2–4
comp05	54	152	9	6	6	139	2–4
comp06	108	361	18	5	5	70	2–4
comp07	131	434	20	5	5	77	2–4
comp08	86	324	18	5	5	61	2–4
comp09	76	279	18	5	5	75	2–4
comp10	115	370	18	5	5	67	2–4
comp11	30	162	5	9	5	13	2–6
comp12	88	218	11	6	6	150	2–4
comp13	82	308	19	5	5	66	2–3
comp14	85	275	17	5	5	60	2–4
comp15	72	251	16	5	5	68	2–4
comp16	108	366	20	5	5	71	2–4
comp17	99	339	17	5	5	70	2–4
comp18	47	138	9	6	6	52	2–3
comp19	74	277	16	5	5	66	2–4
comp20	121	390	19	5	5	78	2–4
comp21	94	327	18	5	5	78	2–4

Fig. 1 The pseudo code for construction algorithm

```
Set population size, Popsize
Set Solution_counter ← 0;
Set Event_counter ← 0;
do while (Solution_counter < Popsize)
      do while(Eventcounter < Number of events)
         Phase 1:Apply largest degree heuristic
      end do
   do while (timetable infeasible)
         Phase 2:Apply neighborhood search
         Phase 3:Apply tabu search
   end do
end do
Return population of feasible timetables
```

the timeslot and the room that satisfy the hard constraints. If the course cannot be scheduled to a specific room, then it will be inserted in any randomly selected room.

In some cases, a feasible timetable is obtained by only employing Phase 1, in which case Phase 2 and Phase 3 are not required. However, feasibility from this Phase is not guaranteed, and Phase 2 and Phase 3 are then employed until feasibility has been achieved.

3.1.2 Phase 2: Neighborhood search

Two neighborhood moves are employed in order to reduce the violations of hard constraints i.e.:

Nbs1: Select a course at random and move to another random feasible timeslot-room.
Nbs2: Select two courses randomly and swap their timeslots and rooms while ensuring feasibility is maintained.

The process stops if there is no improvement on the current timetable after 20 iterations. Here we stop after 20 iterations because we want to give more time for Phase 2 to successfully reach feasible solutions. Note that in Chiarandini et al. (2006) and Landa-Silva and Obit (2008), the process will stop after 10 non improving iterations.

3.1.3 Phase 3: Tabu search

This phase is implemented if the second phase is still not able to generate feasible solutions. Similar neighborhood structures (Nbs1 and Nbs2) are employed with the aim of reducing the time taken in generating feasible timetables. A FIFO structure of tabu list is used in this search. The length of the tabu list, *tl*, is calculated as in Landa-Silva and Obit (2008) i.e. $tl = rand(10) + \delta * nc$ where:

- $rand(10)$ is a random number between 0 and 10.
- nc is the number of events (in the current timetable) that violate the hard constraints.
- δ is a constant which is set to 0.6

This step will terminate after 1000 non-improving iterations.

3.2 An electromagnetism-like mechanism

The main idea of an electromagnetic-like mechanism (EM) that was introduced by Birbil and Fang (2003) stems from electromagnetic-like behavior observed within the field of physics, simulating attraction and repulsion of sample points in order to move towards a promising solution. It begins with a population of randomly generated feasible timetables. The method uses an attraction-repulsion mechanism to move a population of timetables toward optimality. Ideally, the algorithm is based on two particles experiencing forces of mutual attraction or repulsion depending on their individual penalty. The strength of the attraction/repulsion is directly proportional to the product of their charges and inversely proportional to the square of the distance between them. Each particle (in this paper, each particle is considered as a timetable) represents a solution and the charge of each particle relates to its solution quality. The better the solution quality of the particle, the higher charge the particle has. Moreover, the electrostatic force between two point charges is directly proportional to the magnitudes of each charge and inversely proportional to the square of the distance between the charges (see Birbil and Fang 2003). Maenhout and Vanhoucke (2007) presented a novel meta-heuristic technique based on Electromagnetic like mechanism to tackle the nurse scheduling problem. Debels et al. (2006) applied EM algorithm to enhance the movement of a scatter search scheduling algorithm. EM also has been applied successfully by Debels and Vanhoucke (2006) for a project scheduling problem. In our problem, the fixed charge of timetable (particle) i is shown as follows:

$$q^i = \exp\left(-T\frac{f(x^i) - f(x^{best})}{\sum_{k=1}^{m}(f(x^k) - f(x^{best}))}\right) \tag{1}$$

where:

q^i:	the charge for timetable i
$f(x^i)$:	penalty of the timetable i
$f(x^k)$:	penalty of the timetable k
$f(x^{best})$:	penalty of the best timetable
m:	population size
T:	number of timeslots

The solution quality or charge of each individual timetable determines the magnitude of an attraction and repulsion effect in the population. A better solution encourages other particles to converge to attractive valleys while a bad solution discourages particles to move toward this region. These particles move along with the total force and so diversified solutions are generated. The following formulation is the total force, *TF* on particle i:

$$TF_i = \sum_{j\neq i}^{m}\left\{\begin{array}{ll}(f(x^j) - f(x^i))\frac{q^i q^j}{\|f(x^j)-f(x^i)\|^2} & \text{if } f(x^j) < f(x^i) \\ (f(x^i) - f(x^j))\frac{q^i q^j}{\|f(x^j)-f(x^i)\|^2} & \text{if } f(x^j) \geq f(x^i)\end{array}\right\}, \quad \forall i \tag{2}$$

The process of evaluating the total force *TF*, for the course timetabling problem is illustrated in Fig. 2. As is shown resulting timetables 1, 2 and 3 have penalties 210, 165 and 170 respectively. Because Timetable 1 is worse than Timetable 3 while Timetable 2 is better than Timetable 3, Timetable 1 represents a repulsion force which is F_{13} and Timetable 2 encourages Timetable 3 to move to the neighborhood region of Timetable 2 which is an attractive force F_{32}. Consequently, incorporating the search technique outlined in the next section, Timetable 3 moves along with total force *TF* (calculated using the formulas (1) and (2)), to obtain for example Timetable 4.

3.3 A Standard great deluge algorithm

The great deluge algorithm was introduced by Dueck (1993). It is a local search procedure which has certain similarities with simulated annealing by Kirkpatrick et al. (1983) but has been introduced as an alternative. This approach is far less dependent upon parameters than simulated annealing. It needs just two parameters: the amount of computational time that the user wishes to "spend" and an estimate of the quality of solution that a user requires. McMullan and McCollum (2007) proved that

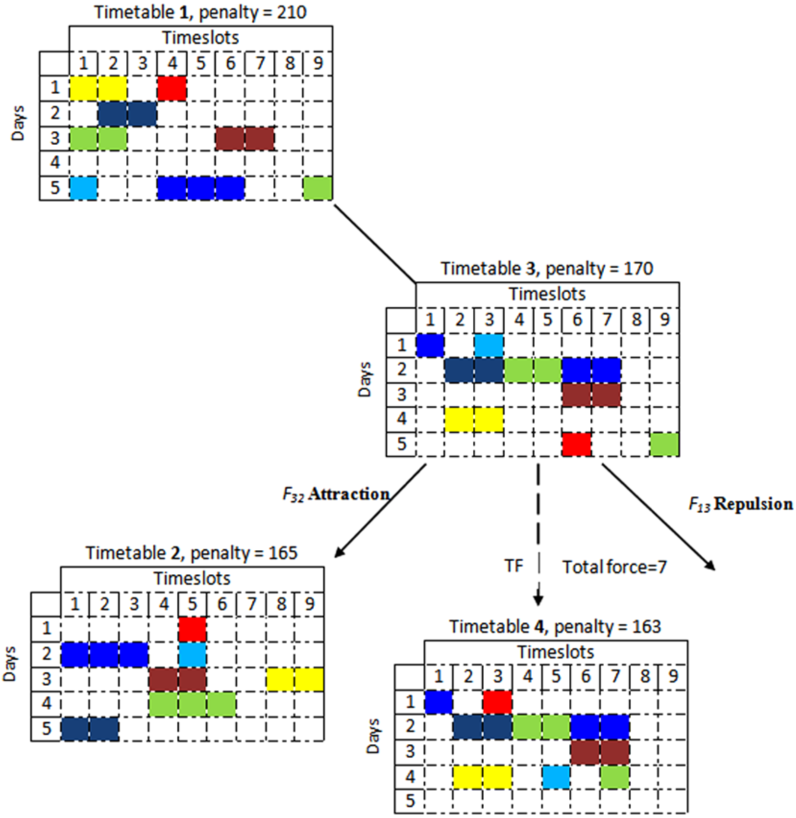

Fig. 2 An example of attract-repulse effect on Timetable 3

the great deluge algorithm is more effective than a simulated annealing algorithm to avoid from being trap in local optima. Apart from accepting a move that improves the solution quality, the great deluge algorithm also accepts a worse solution if the quality of the solution is less than (for the case of minimization) or equal to some given upper boundary value B (in the paper by Dueck it was called a "*level*"). The "*level*" is initially set to be the objective function value of the initial solution. During its run, the "*level*" is iteratively lowered by a constant β where β is a decreasing rate. Examples of research on great deluge algorithms can be found in Burke and Newall (2003) and Petrovic et al. (2007).

3.4 A dynamic force decay rate great deluge based electromagnetic-like mechanism

This algorithm is a combination of two meta-heuristics: (1) electromagnetic-like mechanism and (2) great deluge algorithm. The hybridization is done with the aim of combining the strength of these two algorithms in tackling the two types of course timetabling problem outlined in the previous section. The electromagnetic-like mechanism is used to calculate a total force *TF* and later will be used within the great deluge algorithm to calculate a decreasing rate. This mechanism enables the great

```
Step 1:   Initialization:
          Generate a population of initial solution as Sol_i where i = size of the
          population;
          Calculate the initial penalty cost, f(Sol_i);
          Find the best solution among Sol_i, called Sol_best
          Set total number of iterations, NumOfIte;
          Set number of iteration in great deluge algorithm, NumGD (note that
          NumGD is set to 1000 in our experiments)
          Set iteration ← 0;
          Set iterationGD ← 0;
Step 2:   Evaluation:
          do while (iteration < NumOfIte)
             for each population i where i = 1 to population size;
                Calculate total force, TF_i, based on electromagnetic-like
          mechanism;
                Set initial level: level ← f(Sol_i);
                do while (iterationGD < NumGD || f(Sol_i) < level)
                   Apply a dynamic force decay rate great deluge algorithm;
                end do
             end for;
          end do
```

Fig. 3 Generic pseudo-code for the algorithm

deluge algorithm to act as a multi-start Great Deluge (GD) in which the initial levels are set based on the forces. The fundamental procedures of the algorithm are initialization and evaluation, the later step involves the calculation of the total force and modification of the timetable. The generic pseudo-code for the algorithm is shown in Fig. 3.

The pseudo code for our implementation of the force decay rate great deluge algorithm is adapted from Abdullah and Burke (2006) as presented in Fig. 4.

In Step 1 (see Fig. 3), for every population the quality of the solution Sol_i is measured, called $f(Sol_i)$. At the start, the best solution among Sol_i is set to be Sol_{best}. The initial level is set to be $f(Sol_i)$. In a *do-while* loop (which is in Step 2, Fig. 3), for each population the total force, TF_i, is calculated based on electromagnetic-like mechanism. Note that, in the following explanation, please refer to Fig. 4. This value (TF_i) is used to calculate estimated quality, *EstimatedQuality*, and decay rate, β. A neighborhood structure which randomly assigns courses to a new valid timeslot is applied. This move will generate a new solution, Sol_i^*. The quality of the new solution is measured, $f(Sol_i^*)$. Sol_i^* will be accepted if $f(Sol_i^*)$ is better than the best solution in hand, $f(Sol_{best})$, or if $f(Sol_i^*)$ is less than or equal to the "level". Then, the "level" will be decreased by the value β. The process is repeated and stops when the termination criteria are met (the termination criteria are described for both types of problem in the next section).

Fig. 4 The pseudo code for the dynamic force decay rate great deluge algorithm

```
Calculate estimated quality of a solution,
EstimatedQuality = f(Sol_i) - TF_i;
Calculate force decay rate, β =
EstimatedQuality/NumOfIte;
Define neighborhood of Sol_i by randomly assigning
course to a valid timeslot to generate a new solution
called Sol_i*;
Calculate f(Sol_i*);
if (f(Sol_i*) < f(Sol_best))
      Sol_i ← Sol_i*;
      Sol_best ← Sol_i*;
else
      if (f(Sol_i*) ≤ level)
            Sol_i ← Sol_i*;
level = level - β;
```

4 Experiments and results

The proposed algorithm was programmed using Matlab and simulations were performed on an Intel Pentium 4 2.33 GHz computer and tested on eleven standard benchmark course timetabling problem and twenty one datasets on curriculum-based course timetabling problems. The parameters used were chosen after preliminary experiments. The population size is set to 50, and is comparable to similar experimentation in the literature (Birbil and Fang 2003).

4.1 Post enrollment-based course timetabling problem

The first series of experiments carried out in this section minimize the number of students that has a course scheduled in the last timeslot of the day, a student that has more than 2 consecutive courses and a student that has a single course on a day. The following subsections illustrate two cases of the post-enrollment course timetabling problems i.e. (i) Socha datasets and (ii) ITC2002 datasets.

4.1.1 Socha datasets

The details of these results can also be found in Turabieh et al. (2009). Termination is based on number of generations, and is initially set at 100,000 iterations. We compare our approach with other algorithms on the eleven timetabling instances. The algorithms compared in the table are described as follows:

- M1: The genetic algorithm and local search by Abdullah and Turabieh (2008). They tested a genetic algorithm with a repair function and local search on course timetabling problems.
- M2: The randomized iterative improvement algorithm by Abdullah et al. (2007a). They presented composite neighborhood structures with a randomized iterative improvement algorithm.

M3: The graph hyper heuristic by Burke et al. (2007). They employed tabu search with graph-based hyper-heuristics for the course and exam timetabling problems.
M4: The variable neighborhood search with tabu by Abdullah et al. (2005). They used a variable neighborhood search based on a random descent local search with Monte-Carlo acceptance criterion.
M5: The hybrid evolutionary approach by Abdullah et al. (2007b). They used a randomized iterative improvement algorithm as a local search with a mutation operator.
M6: The extended great deluge by McMullan (2007).
M7: The non linear great deluge by Linda-Silva and Obit (2008).
M8: The local search method by Socha et al. (2002).
M9: The ant algorithm by Socha et al. (2002). They developed an ant colony optimization algorithm with a construction graph model.
M10: The fuzzy algorithm by Asmuni et al. (2005). They focused on fuzzy based methods in ordering three different heuristics.
M11: The evolutionary algorithm by Rossi-Doria et al. (2003).

The best results out of 5 runs obtained are presented. Table 4 shows the comparison of our results in terms of penalty cost with other available approaches in the literature on eleven instances. The term "x%Inf." in Table 4 indicates a percentage of runs that failed to obtain feasible solutions. The best results are presented in bold. Our algorithm is capable of finding feasible timetables for all cases.

It can be seen that the extended great deluge by McMullan (2007) has better results compared to others, followed by non-linear great deluge by Landa-Silva and Obit (2008). In general, our approach is able to obtain competitive results with other approaches in the literature. We extended our experiments by increasing the number of iterations (200,000 iterations) with the objective of demonstrating that our algorithm is able to produce good results given extra processing time. We note that in real world situations, course timetabling is an off line problem, and the processing time is usually not critical. The emphasis in this paper is on generating good quality solutions, the price to pay for this is that there may be required an extended amount of computational time. In a real world situation, course timetables are created months before they are required, thus finding a feasible solution is more important than prolonging the search in order to generate good solutions. Table 5 shows the comparison of our approach by prolonging the computational time with best known results in the literature. The average running time for small datasets is 90 seconds, medium datasets is 2 hours and large dataset is 6 hours. The time taken here is considered acceptable based on the complexity of each dataset. We use the same amount of iterations i.e. 200,000 as Landa-Silva and Obit (2008) (note that the authors set a different number of iterations for different groups of datasets) and McMullan (2007). Note that only medium and large datasets are considered in this extended experiment. The small datasets are not considered here due to the ability of our approach in obtaining global optima in the previous experiment (see Table 4).

Again, the best results are presented in bold. Our objective here is to show that our approach is not only able to produce good quality solutions but improves on the best known results on four datasets. The extended experiments are able to improve

Table 4 Comparison of our results with other approaches in the literature

Dataset	Our method	M1	M2	M3	M4	M5
*small*1	**0**	2	**0**	6	**0**	**0**
small 2	**0**	4	**0**	7	**0**	**0**
small 3	**0**	2	**0**	3	**0**	**0**
small 4	**0**	**0**	**0**	3	**0**	**0**
small 5	**0**	4	**0**	4	**0**	**0**
*medium*1	175	254	242	372	317	221
*medium*2	197	258	161	419	313	147
*medium*3	216	251	265	359	357	246
*medium*4	149	321	181	348	247	165
*medium*5	190	276	151	171	292	130
large	912	1026	100% Inf	1068	100% Inf	**529**

Dataset	M6	M7	M8	M9	M10	M11
*small*1	**0**	3	8	1	10	**0**
small 2	**0**	4	11	3	9	3
small 3	**0**	6	8	1	7	**0**
small 4	**0**	6	7	1	17	**0**
small 5	**0**	**0**	5	**0**	7	**0**
*medium*1	**80**	140	199	195	243	280
*medium*2	**105**	130	202.5	184	325	188
*medium*3	**139**	189	77.5% Inf	248	249	249
*medium*4	**88**	112	177.5	164.5	285	247
*medium*5	**88**	141	100% Inf	219.5	132	232
large	730	876	100% Inf	851.5	1138	100%Inf

Table 5 Comparison with best known results

Dataset	Our approach		% improve with	Best
	100K iterations	200K iterations	long run	known
*medium*1	175	96	45.14%	**80**
*medium*2	197	**96**	51.27%	105
*medium*3	216	**135**	37.50%	139
*medium*4	149	**79**	46.98%	88
*medium*5	190	**87**	54.21%	88
large	912	683	25.11%	**529**

the solutions between 25% to 54% compared to our previous results. This illustrates the effectiveness of our approach given extra computational time. It is interesting to find that the large dataset obtains the lowest improvement i.e. 25.11% compared

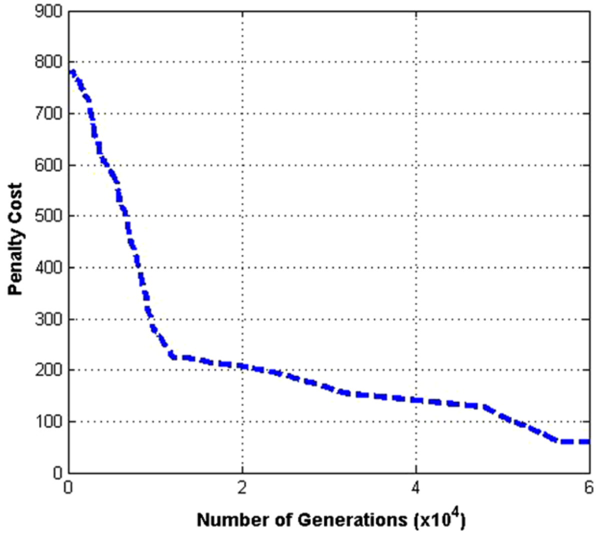

Fig. 5 The result of the algorithm applied on *medium*4 dataset

to medium datasets. It can be argued that a higher number of courses, students and room feature requirements (see Table 1) would imply that we might have less of and more sparsely distributed solution points (feasible solutions) in our solution space given the hard constraints which must be satisfied. However, our approach shows an impressive improvement for medium datasets in which we believe that there are much more solution points in the solution space even given the same amount of courses (and where the number of students, M, and the requirement of room features, F, are less). This shows that these two parameters (i.e. M and F) are significant factors that influence the search process in exploring the search space.

Figure 5 shows the performance of our approach on the *medium*4 dataset. This graph demonstrates how the algorithm explores the search space. The x-axis represents the number of iterations whilst the y-axis represents the penalty cost. The curve shows that the algorithm begins with an initial solution and rapidly improves the results in less than 10,000 iterations. It is believed the quality of the solutions obtained in these experiments can be attributed to the ability of the algorithm in effective exploration of different regions of the solution space, applied to 50 different solutions for each iteration. The figure also shows that by prolonging the search process, our approach is able to further improve resultant solutions. However, the longer the search times, the slower the rate of improvement.

4.1.2 ITC2002 datasets

The second experiments on the Post-enrollment course timetabling problems are performed based on ITC2002 datasets where all the hard and soft constraints are same as in Socha datasets. Table 6 shows the comparison of our final results in terms of penalty cost (out of 5 runs) compared to other recent published results taken from Chiarandini et al. (2006). The execution time is based on ITC2002 rules i.e. 547 seconds for each instance.

Table 6 Comparison results on ITC2002

Instance	1	2	3	4	5	6	7	8	9	10
Official winner	**45**	25	65	115	102	13	44	29	17	61
Best known result	**45**	**14**	**45**	**71**	**59**	**1**	**3**	**1**	**8**	**52**
Our results	52	20	78	74	71	6	6	15	32	58
Instance	11	12	13	14	15	16	17	18	19	20
Official winner	44	107	78	52	24	22	86	31	44	7
Best known result	**30**	**75**	**55**	**18**	**8**	**5**	**46**	**24**	**33**	**0**
Our results	**30**	88	105	51	34	10	121	26	57	5

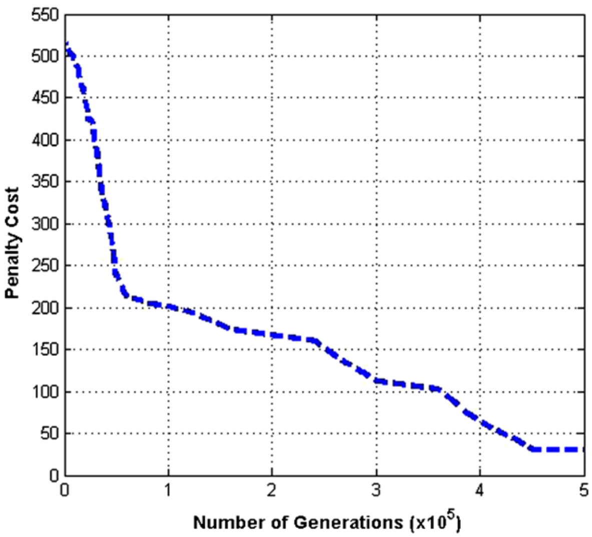

Fig. 6 The result of the algorithm applied on *instance* 11

From Table 6, we can see that our approach is able to obtain feasible solutions for all datasets. We obtained better results on 13 instances than the official winner in ITC2002, and ties on one instance with the best known result (i.e. *instance* 11).

Figure 6 shows the performance of our approach on *instance* 11. The x-axis represents the number of iterations whilst the y-axis represents the penalty cost. Again, this graph represents how the algorithm explores the search space. The analysis of the graph shows that there is a trend of the cost improvement as the number of generation increases. The slope of the curve indicates a small decrease in the penalty cost as the number of generation increases. It offers a high improvement at the early stage of the search space.

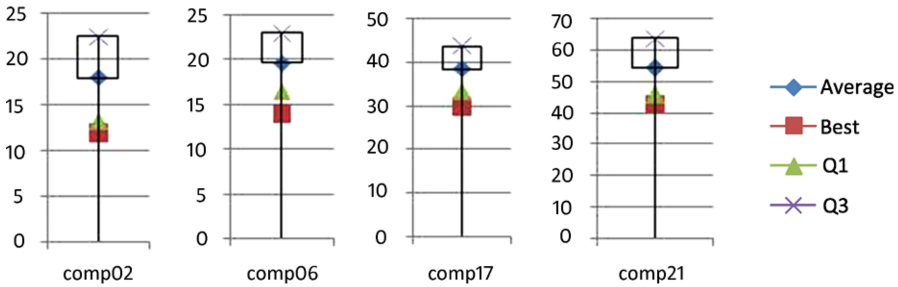

Fig. 7 Box plots of penalty costs

4.2 Curriculum-based course timetabling problem

The second series of experiments in this section deals with room capacity, minimum working days, isolated lectures and room stability as stated in Sect. 2.2. The following subsections illustrate two cases of the Curriculum-based Course Timetabling Problem; Basic Formulation (UD1) and ITC2007 Formulation (UD2).

4.2.1 Basic formulation (UD1)

The Basic formulation of the Curriculum-based Course Timetabling Problem was released in 2006 by Gaspero et al. (2007), in which all hard constraints and three out of four soft constraints i.e. S1, S2 and S3 (see Sect. 2.2) are considered. Table 7 shows the results obtained and the comparison with best known solutions.

The termination condition for each run is set to 600 seconds which is based on the time allocated in the ITC2007.

The best results, average and standard deviation out of 10 runs are shown in Table 7 with varying random seeds. From Table 7, we can see that our approach is able to obtain better or equal results on eight instances compared to best known results in the literature.

Figure 7 shows the box plot of the penalty cost on some of the instances considered in this experiment. The results from the figures show less dispersions of solution points.

4.2.2 ITC2007 formulation (UD2)

The second experiments on Curriculum-based Course Timetabling Problem are performed based on ITC2007 constraints: all hard and soft constraints are considered (see Sect. 2.2). Table 6 shows the comparison of our final results in terms of penalty cost compared to other recent published results in literature. Each run takes 600 seconds for both construction and improvement phases based on our machine hardware (as mentioned in Sect. 4) to make a reasonable comparison between other results. The algorithms under comparison in the table comprise experiments tested during the ITC2007 and other experiments under a greater computational time, and are described as follows:

Table 7 Comparison results on curriculum-based course timetabling-UD1

Instance	Initial Solution	Our Approach				Best known Solution[a]	
		Best	Avg.	Std Dev	Time (s)	Results	Method Used
comp01	1869	**4**	5.9	2.28	346.85	**4**	Tabu Search
comp02	6776	**12**	18	4.69	462.81	20	Tabu Search
comp03	6041	40	46.1	4.17	372.21	38	Tabu Search
comp04	4429	23	28	4.42	299.61	18	Tabu Search
comp05	7513	230	237.4	6.25	173.25	219	Tabu Search
comp06	4310	**14**	19.7	3.33	267.46	16	Mathematical Programming
comp07	3119	10	17.6	4.37	422.36	3	Mathematical Programming
comp08	3007	25	29.4	3.37	530.20	20	Mathematical Programming
comp09	4537	65	73.9	6.04	493.06	54	Tabu Search
comp10	2479	10	13.1	3.28	236.80	2	Mathematical Programming
comp11	1212	**0**	0	0	155.42	**0**	Tabu Search
comp12	3155	252	265.2	10.95	142.25	239	Tabu Search
comp13	4828	40	47.8	5.90	537.96	32	Tabu Search
comp14	3254	33	39.3	4.47	583.32	27	Tabu Search
comp15	5717	39	41.4	2.67	534.82	28	Tabu Search
comp16	4888	**11**	19.7	6.66	428.27	16	Tabu Search
comp17	3808	**30**	38.6	5.54	455.78	34	Tabu Search
comp18	1495	38	46.5	5.85	483.71	34	Tabu Search
comp19	4609	37	44.1	5.62	338.72	32	Tabu Search
comp20	5852	**2**	8.3	5.01	399.59	11	Tabu Search
comp21	4459	**43**	54.5	8.94	367.91	52	Tabu Search

[a] http://tabu.diegm.uniud.it/ctt/index.php

A1: The repair-based timetable solver by Clark et al. (2008)
A2: The threshold acceptance metaheuristic by Geiger (2008)
A3: The tabu search approach by Lü and Hao (2010)
A4: The constraint-based solver by Müller (2009)
A5: The constraint satisfaction problem by Atsuta et al. (2007)
A6: The dynamic tabu search by De Cesco et al. (2008)
A7: The integer programming method by Lach and Lübbecke (2008)
A8: The tabu search with relaxed stopping condition by Schaerf (in Lü and Hao 2010)

Again, the best results out of 5 runs obtained are presented. Table 8 shows the comparison of our results in terms of penalty cost with other available approaches in the literature on twenty one instances. Note that the shaded areas represent results from ITC2007. Other results were obtained under a different number of trials but same computational time. The term "–" in Table 8 indicates that the instances have not been attempted in the experiment. The best results are presented in bold.

From Table 8, we can see that our approach is able to obtain better or equal results on four instances compared to best known results in the literature. In all of

Table 8 Comparison results on curriculum-based course timetabling

Instance	Our approach Best	Ave	A1	A2	A3	A4	A5	A6	A7	A8	Best[a] known
comp01	**5**	5	9	**5**	**5**	**5**	**5**	5	13	**5**	**5**
comp02	39	53.9	103	108	34	43	50	75	43	56	**29**
comp03	76	84.2	101	115	70	72	82	93	76	79	**66**
comp04	**35**	51.9	55	67	38	35	**35**	45	38	38	**35**
comp05	315	339.5	370	408	298	298	312	326	314	316	**292**
comp06	50	64.4	112	94	47	41	69	62	41	55	**28**
comp07	12	20.2	97	56	19	14	42	38	19	26	**6**
comp08	**37**	47.9	72	75	43	39	40	50	43	42	38
comp09	104	113.9	132	153	99	103	110	119	102	104	**96**
comp10	10	24.1	74	66	16	9	27	27	14	19	**4**
comp11	**0**	0	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
comp12	337	355.9	393	430	320	331	351	358	405	342	**310**
comp13	61	72.4	97	101	65	66	68	77	68	72	**59**
comp14	53	63.3	87	88	52	53	59	59	54	57	**51**
comp15	73	88	119	128	69	84	82	87	–	79	**68**
comp16	32	51.7	84	81	38	34	40	47	–	46	**22**
comp17	72	86.2	152	124	80	83	102	86	–	88	**60**
comp18	77	85.8	110	116	67	83	68	71	–	75	**65**
comp19	60	78.1	111	107	59	62	75	74	–	64	**57**
comp20	22	42.9	144	88	35	27	61	54	–	32	**4**
comp21	95	121.5	169	174	105	103	123	117	–	107	**86**

[a] http://tabu.diegm.uniud.it/ctt/index.php

the cases, our approach is better than Clark et al. (2008), and Geiger (2008) (equal on comp01 and comp11), better than Atsuta et al. (2007) (equal on comp01, comp04 and comp11) and Müller (2009) (equal on comp01 and comp11) on seventeen instances, and better than Lü and Hao (2010) on fifteen instances (also equal on comp01 and comp11). Figure 8 shows the box plot of the penalty cost on some of the instances considered in this experiment. The results from the figures show less dispersions of solution points.

Additional experiments have been carried out to test the performance of our approach on additional instances coded as DDS1–DDS7, Test1–Test4 and Toy. The details of these instances such as the data sets, solution checker, portfolio of formulations for the curriculum-based course timetabling, and other solutions contributed from the research community can be found at http://tabu.diegm.uniud.it/ctt/index.php. The results from Table 9 show that our approach is able to obtain optimal solutions on DDS2–DDS7 instances and able to beat best known results on the Test3 instance.

We believe that introducing a dynamic force value that is iteratively calculated using EM algorithm (treated as a decreasing rate) helps the great deluge algorithm in accepting or rejecting a new solution during the search process, rather than using a fixed decreasing value as originally proposed in the standard great deluge algorithm.

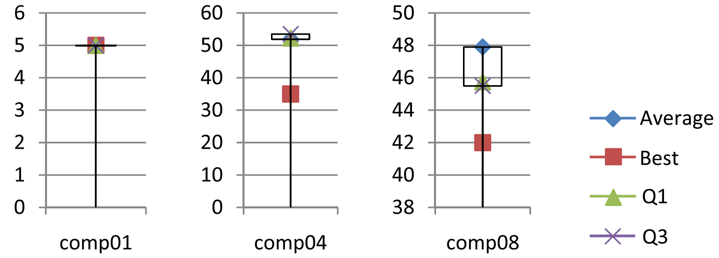

Fig. 8 Box plots of penalty costs

Table 9 Comparison with best known results on additional course timetabling instances

Dataset	DDS1	DDS2	DDS3	DDS4	DDS5	DDS6	DDS7
Our approach	155	**0**	**0**	**30**	**0**	**4**	**0**
Best known[a]	**83**	**0**	**0**	**30**	**0**	**0**	**0**

Dataset	Test1	Test2	Test3	Test4	Toy
Our approach	**224**	**16**	**67**	82	**0**
Best known[a]	**224**	**16**	73	**73**	**0**

[a]http://tabu.diegm.uniud.it/ctt/index.php

It also helps the great deluge algorithm to control the estimated value rather than manually set (as in standard great deluge). This mechanism helps the search process to explore more points in different directions. This can increase the possibility of obtaining improved solutions.

5 Conclusion

This paper presents a force decay rate great deluge based electromagnetism-like mechanism to solve the course timetabling problems, where a force calculated from the electromagnetic-like mechanism is applied as a decreasing rate to be used within the great deluge algorithm. To our knowledge, this is the first such algorithm aimed at this problem domain. In order to test the performance of our approach, experiments are carried out based on enrollment-based and curriculum-based benchmark problems and compared with state-of-the-art methods from the literature. The experimental results show that the proposed approach is competitive and works comparatively well across all problem instances in comparison with other approaches studied in the literature. The fact that the approach provides good solutions to two different models of the UCTP points to the generality of the approach. With the help of the dynamic force value, it is clear that our approach is effective in finding (near) optimal solutions for the course timetabling problem and hence can act as one of the powerful tools in solving difficult problems within this domain.

References

Abdullah, S., Burke, E.K.: A multi-start large neighborhood search approach with local search methods for examination timetabling. In: The International Conference on Automated Planning and Scheduling (ICAPS 2006), pp. 334–337 (2006)

Abdullah, S., Turabieh, H.: Generating university course timetable using genetic algorithm and local search. In: Proceeding of the 3rd International Conference on Hybrid Information Technology, pp. 254–260 (2008)

Abdullah, S., Burke, E.K., McCollum, B.: An investigation of variable neighborhood search for university course timetabling. In: The 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA), pp. 413–427 (2005)

Abdullah, S., Burke, E.K., McCollum, B.: Using a randomized iterative improvement algorithm with composite neighborhood structures for university course timetabling. In: Metaheuristics: Progress in Complex Systems Optimisation. Operations Research / Computer Science Interfaces Series. Springer, Berlin (2007a). Chap. 8. ISBN: 978-0-387-71919-1

Abdullah, S., Burke, E.K., McCollum, B.: A hybrid evolutionary approach to the university course timetabling problem. In: IEEE Congress on Evolutionary Computation, pp. 1764–1768 (2007b). ISBN: 1-4244-1340-0

Asmuni, H., Burke, E.K., Garibaldi, J.M.: Fuzzy multiple heuristic ordering for course timetabling. In: The Proceedings of the 5th United Kingdom Workshop on Computational Intelligence (UKCI05), London, UK, pp. 302–309 (2005)

Atsuta, M., Nonobe, K., Ibaraki, T.: ITC-2007 Track2: An Approach using General CSP Solver (2007). http://www.cs.qub.ac.uk/itc2007/winner/bestcoursesolutions/Atsuta_et_al.pdf

Bardadym, VA: Computer-aided school and university timetabling: A new wave. In: Burke, E.K., Ross, P. (eds.) Practice and Theory of Automated Timetabling V. LNCS, vol. 1153, pp. 22–45. Springer, Berlin (1996)

Birbil, S.I., Fang, S.C.: An electromagnetism-like mechanism for global optimization. J. Glob. Optim. **25**, 263–282 (2003)

Burke, E.K., Newall, J.P.: Enhancing timetable solutions with local search methods. In: Burke, E.K., De Causmaecker, P. (eds.) Selected Papers from 4th International Conference on the Practice and Theory of Automated Timetabling. LNCS, vol. 2740, pp. 195–206. Springer, Berlin (2003)

Burke, E.K., Petrovic, S.: Recent research directions in automated timetabling. Eur. J. Oper. Res. **140**(2), 266–280 (2002)

Burke, E.K., Elliman, D.G., Ford, P.H., Weare, R.F.: Examination timetabling in British universities—A survey. In: Burke, E.K., Ross, P. (eds.) Selected Papers from 1st International Conference on the Practice and Theory of Automated Timetabling. LNCS, vol. 1153, pp. 76–92 (1996)

Burke, E.K., Kendall, G., Soubeiga, E.: A tabu-search hyper-heuristic for timetabling and rostering. J. Heuristics **9**(6), 451–470 (2003)

Burke, E.K., Meisels, A., Petrovic, S., Qu, R.: A graph-based hyper-heuristic for timetabling problems. Eur. J. Oper. Res. **176**, 177–192 (2007)

Burke, E.K., Marecek, J., Parkes, A., Rudová, H.: Decomposition, reformulation, and diving in university course timetabling. Comput. Oper. Res. **37**(3), 582–597 (2009)

Carter, M.W.: A survey of practical applications of examination timetabling algorithms. Oper. Res. **34**(2), 193–202 (1986)

Carter, M.W., Laporte, G.: Recent developments in practical examination timetabling. In: Burke, E.K., Ross, P. (eds.) LNCS, vol. 1153, pp. 373–383 (1996)

Chiarandini, M., Birattari, M., Socha, K., Rossi-Doria, O.: An effective hybrid algorithm for university course timetabling. J. Sched. **9**(5), 403–432 (2006)

Clark, M., Henz, M., Love, B.: QuikFix. A repair-based timetable solver. In: Proceedings of the Seventh PATAT Conference (2008). http://www.comp.nus.edu.sg/~henz/publications/ps/PATAT2008.pdf

De Cesco, F., Di Gaspero, L., Schaerf, A.: Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, and results. In: Proceedings of the Seventh PATAT Conference (2008). http://tabu.diegm.uniud.it/ctt/DDS2008.pdf

de Werra, D.: An introduction to timetabling. Eur. J. Oper. Res. **19**(2), 151–162 (1985)

Debels, D., Vanhoucke, M.: An electromagnetism meta-heuristic for the resource-constrained project scheduling problem. In: LNCS, vol. 3871, pp. 259–270. Springer, Berlin (2006)

Debels, D., De Reyck, B., Leus, R., Vanhoucke, M.: A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. Eur. J. Oper. Res. **169**(2), 638–653 (2006)

Di Gaspero, L., Schaerf, A., McCollum, B.: The Second International Timetabling Competition (ICT-2007) Curriculum-based Course Timetabling (Track 3) (2007). http://pst.istc.cnr.it/ssc-at-icaps-07/papers/digaspero-et-al-SSC07.pdf

Dueck, G.: New Optimization Heuristics. The great deluge algorithm and the record-to-record travel. J. Comput. Phys. **104**, 86–92 (1993)

Geiger, M.J.: An application of the threshold accepting metaheuristic for curriculum based course timetabling. In: Proceedings of the Seventh PATAT Conference (2008). arXiv:0809.0757v1

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimisation by simulated annealing. Science **220**, 671–380 (1983)

Lach, G., Lübbecke, M.E.: Curriculum based course timetabling: Optimal solutions to the Udine benchmark instances. In: Proceedings of the Seventh PATAT Conference (2008). http://www.math.tu-berlin.de/luebbeck/papers/udine.pdf

Landa-Silva, D., Obit, J.H.: Great deluge with non-linear decay rate for solving course timetabling problem. In: The Fourth International IEEE Conference on Intelligent Systems, Varna, Bulgaria, pp. 11–18 (2008)

Lü, Z., Hao, J.-K.: Adaptive Tabu Search for course timetabling. Eur. J. Oper. Res. **200**(1), 235–244 (2010)

Maenhout, B., Vanhoucke, M.: An electromagnetic meta-heuristic for the nurse scheduling problem. J. Heuristics **13**, 315–401 (2007)

McCollum, B.: A perspective on bridging the gap between theory and practice in university timetabling. In: LNCS, vol. 3867, pp. 3–23. Springer, Berlin (2007)

McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A., Di Gaspero, L., Qu, R., Burke, E.K.: Setting the research agenda in automated timetabling: The second international timetabling competition. INFORMS J. Comput. **22**(1), 120–130 (2010)

McMullan, P.: An extended implementation of the great deluge algorithm for course timetabling. In: Computational Science—ICCS, Part I. LNCS, vol. 4487, pp. 538–545. Springer, Berlin (2007)

McMullan, P., McCollum, B.: Dynamic job scheduling on the grid environment using the great deluge algorithm. In: The 9th International Conference on Parallel Computing Technologies. LNCS, vol. 4671, pp. 283–292. Springer, Berlin (2007)

Müller, T., ITC2007: Solver description. In: Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling (2007). http://www.unitime.org/papers/itc2007.pdf

Müller, T., ITC2007 solver description: a hybrid approach. Ann. Oper. Res. **172**(1), 429–446 (2009)

Petrovic, S., Burke, E.K.: University timetabling. In: Leung, J. (ed.) Handbook of Scheduling: Algorithms, Models, and Performance Analysis. CRC Press, Boca Raton (2004). Chap. 45

Petrovic, S., Yang, Y., Dror, M.: Case-based selection of initialisation heuristics for metaheuristic examination timetabling. Expert Syst. Appl. **33**(3), 772–785 (2007)

Rossi-Doria, O., Samples, M., Birattari, M., Chiarandini, M., Dorigo, M., Gambardella, L.M., Knowles, J., Manfrin, M., Mastrolilli, M., Paechter, B., Paquete, L., Stützle, T.: A comparison of the performance of different meta-heuristics on the timetabling problem. In: Burke, E.K., De Causmaecker, P. (eds.) The Practice and Theory of Automated Timetabling IV: Selected Papers from 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT IV), Gent, Belgium. LNCS, vol. 2740, pp. 329–354. Springer, Berlin (2003)

Schaerf, A.: A survey of automated timetabling. Artif. Intell. Rev. **13**(2), 87–127 (1999)

Socha, K., Knowles, J., Samples, M.: A max-min ant system for the university course timetabling problem. In: The Proceedings of the 3rd International Workshop on Ant Algorithms (ANTS 2002). LNCS, vol. 2463, pp. 1–13. Springer, Berlin (2002)

Turabieh, H., Abdullah, S., McCollum, B.: Electromagnetism-like mechanism with force decay rate great deluge for the course timetabling problem. In: The Fourth International Conference on Rough Set and Knowledge Technology (RSKT 2009). LNAI, vol. 5589, pp. 497–504. Springer, Berlin (2009)